AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) In a local server that receives data from one or more remote entities over a data transport protocol, a method of applying a cryptographically secure hash to packets from unverified remote entities for preventing denial of service attacks on lookup tables used to store state information for one or more remote entities, while maintaining the performance of the local server for packets from verified remote entities, the method comprising the acts of:

~~(A)~~ receiving a packet of data from a remote entity that includes connection identifier information;

~~(B)~~ hashing at least a portion of the connection identifier information using a first hash function to generate a first hash, the first hash identifying an entry in a first table of verified remote entities, the first table for storing remote entities that have a confirmed connection identifier; ~~the entry containing state information for all packets comprising the first hash; and~~

~~(C)~~ determining [[if]]when state information for the remote entity exists at the entry in the first table of verified remote entities~~;(a) wherein if it is determined the state information for the remote entity does exist in the first table of verified remote entities~~, and performing standard data transport protocol on the packet of data; and

~~(b) providing program modules for performing the following~~ determining when ~~it is determined that the~~ state information for the remote entity does not exist in the first table of verified remote entities and performing the following:

~~(i)~~ hashing at least a portion of the connection identifier information using a second hash function that is ~~more computationally intensive and more~~ cryptographically secure ~~than the first hash function~~ to generate a second hash, the second hash function having a lower probability than the first hash function of generating an identical hash for connection identifier information from more than one remote entity~~resulting in a second hash which is less predictable than the first hash generated by the first hash function~~, the second hash identifying ~~another~~ a second entry in a second table of unverified remote entities, the second table for storing remote entities that do not have a confirmed connection identifier; ~~the second entry containing state information for all packets comprising the second hash; and~~

~~(ii)~~ determining [[if]]when state information for the remote entity exists at the second entry in the second table of unverified remote entities~~; (1) wherein if it is determined that the state information for the remote entity exists in the second table of unverified remote entities~~, and comparing secret information provided within the packet of data with information previously supplied to the remote entity for determining if the remote entity can be verified such that state information can be moved to the first table of verified remote entities; and

~~(2)~~ determining when ~~wherein if it is determined that the~~ state information for the remote entity does not exist in the second table of unverified remote entities[[;]] and checking whether the local server is a listener that may accept the packet of data from the remote entity for determining [[if]]when the state information for the remote entity should be created in the second table of unverified remote entities.

2. (Cancelled).

3. (Previously Presented) The method of claim 1, wherein the standard data transport protocol is transmission control protocol.

4. (Currently Amended) The method of claim 1, wherein ~~if the~~ when state information for the remote entity exists in the second table of unverified remote entities, but the remote entity cannot be verified, the method further comprises the act of:

checking if the packet includes a synchronization message for determining how to respond to the unverified remote entity.

5. (Original) The method of the claim 4, wherein if the packet of data includes a synchronization message, the local server responds by either sending a synchronization-acknowledgement packet or by deleting the packet.

6. (Previously Presented) The method of the claim 4, wherein if the packet of data does not include a synchronization message, the local server responds by one or more of the following: deleting the packet, retransmitting the original message to the remote entity or removing the state information from the second table of unverified remote entities.

7. (Previously Presented) The method of claim 1, wherein the first hash function is also a cryptographically secured hash function.

8. (Original) The method of claim 7, wherein the first and second hash functions are one of hardware based or software based.

9. (Currently Amended) The method of claim 1, wherein [if]when state information for the remote entity does not exist in either the first table of verified remote entities or the second table of unverified remote entities, and wherein the server is a listener that may accept the package of data from the remote entity, the method further comprising the acts of:

creating state information for the remote entity within the second table of unverified remote entities; and

sending a synchronization-acknowledgement packet that includes an initial sequence number to the remote entity.

10. (Currently Amended) The method of claim 1, wherein [[if]]when state information for the remote entity does not exist in either the first table of verified entities or the second table of

unverified entities, and the server is not a listener that may accept the package of data from the remote entity, the method further comprises the act of:

sending a reset command to the remote entity for indicating that the packet was not verifiable and needs to be resent.

11. (Previously Presented) The method of claim 1, wherein the remote entity becomes verified by sharing a secret sent to the remote entity by the local server.

12. (Currently Amended) In a local server that receives data from one or more remote entities over a data transport protocol, a method of applying a cryptographically secure hash to packets from unverified remote entities for preventing denial of service attacks on lookup tables used to store state information for one or more remote entities, while maintaining the performance of the local server for packets from verified remote entities, the method comprising:

an act of receiving a packet of data from a remote entity that includes connection identifier information;

an act of hashing at least a portion of the connection identifier information using a first hash function to generate a first hash, the first hash identifying an entry in a first table of verified remote entities, the first table for storing remote entities that have a confirmed connection identifier;

a step for determining [[if]]when state information exists for the remote entity in [[a]]the first table of verified remote entities;

[[if]]when the state information for the remote entity does not exist in the first table of verified remote entities, performing the following:

an act of hashing at least a portion of the connection identifier information using a second hash function that is cryptographically secure to generate a second hash, the second hash function having a lower probability than the first hash function of generating an identical hash for connection identifier information from more than one remote entity, the second hash identifying a second entry in a second table of unverified remote entities, the second table for storing remote entities that do not have a confirmed connection identifier;

a step for determining [[if]]when state information exists for the remote entity in [[a]]the second table of unverified remote entities;

[[if]]when the state information exists in the second table of unverified remote entities, a step for determining [[if]]when the remote entity can be verified such that state information can be moved to the first table of verified remote entities; and [[if]]when state information does not exist in the second table of unverified remote entities; a step for determining [[if]]when state information for the remote entity should be created in the second table of unverified remote entities.

13. (Currently Amended) The method of claim 12, wherein [[if]]when the state information for the remote entity does exist in the first table of verified remote entities, standard data transport protocol processing is performed.

14. (Original) The method of claim 13, wherein the standard data transport protocol is transmission control protocol.

15. (Currently Amended) The method of claim 12, wherein [[if]]when the state information exists in the second table of unverified remote entities, but the remote entity cannot be verified, the method further comprises the act of:

checking if the packet includes a synchronization message for determining how to respond to the unverified remote entity.

16. (Original) The method of the claim 15, wherein if the packet of data includes a synchronization message, the local server responds by either sending a synchronization-acknowledgement packet or by deleting the packet.

17. (Previously Presented) The method of the claim 15, wherein if the packet of data does not include a synchronization message, the local server responds by one or more of the following: deleting the packet, retransmitting the original message to the remote entity or removing the state information from the second table of unverified remote entities.

18. (Cancelled)

19. (Original) The method of claim 18, wherein the first hash function is also a cryptographically secured hash function, and wherein the first and second hash functions are one of either hardware based or software based.

20. (Currently Amended) The method of claim 12, wherein [[if]]when state information does not exist in either the first table of verified remote entities or the second table of unverified remote entities, and wherein the step for determining [[if]]when state information for the remote entity should be created in the second table of unverified remote entities includes the act of checking whether the local server is a listener that may accept the package of data from the remote entity, the method further comprising the acts of:

creating state information for the remote entity within the second table of unverified remote entities; and

sending a synchronization-acknowledgement packet that includes an initial sequence number to the remote entity.

21. (Currently Amended) The method of claim 12, wherein [[if]]when state information does not exist in either the first table of verified entities or the second table of unverified entities, the step for determining if state information for the remote entity should be created in the second table of unverified remote entities includes the act of checking whether the local server is a listener that may accept the package of data from the remote entity, and the server is not a listener that may accept the package of data from the remote entity, the method further comprising the act of:

sending a reset command to the remote entity for indicating that the packet was not verifiable and needs to be resent.

22. (Currently Amended) For a local server that receives data from one or more remote entities over a data transport protocol, a computer program product comprising computer readable storage media storing computer executable instructions that implement the method recited in claim 1. ~~a method of applying a cryptographically secure hash to packets from unverified remote entities for preventing denial of service attacks on lookup tables used to store state information for one or more remote entities, while maintaining the performance of the local server for packets from verified remote entities, the method comprising the acts of:~~

~~(A) receiving a packet of data from a remote entity that includes connection identifier information;~~

~~(B) hashing at least a portion of the connection identifier information using a first hash function to generate a first hash, the first hash identifying an entry in a first table of verified remote entities, the entry containing state information for all packets comprising the first hash; and~~

~~(C) determining if state information for the remote entity exists at the entry in the first table of verified remote entities;~~

~~(a) wherein if it is determined that the state information for the remote entity does exist in the first table of verified remote entities, performing standard data transport protocol on the packet of data; and~~

~~(b) providing program modules for performing the following when it is determined that the state information for the remote entity does not exist in the first table of verified remote entities:~~

~~(i) hashing at least a portion of the connection identifier information using a second hash function that is more computationally intensive and more cryptographically secure than the first hash function, resulting in a second hash which is less predictable than the first hash generated by the first hash function, the second hash identifying an entry in a second table of unverified remote entities, the second entry containing state information for all packets comprising the second hash; and~~

~~(ii) determining if state information for the remote entity exists at the second entry in the table of unverified remote entities;~~

~~(1) wherein if it is determined that the state information for the remote entity exists in the second table of unverified remote entities, comparing secret information provided within the packet of data with information previously supplied to the remote entity for determining if the remote entity can be verified such that state information can be moved to the first table of verified remote entities; and~~

~~(2) wherein if it is determined that the state information for the remote entity does not exist in the second table of unverified remote entities; checking whether the local server is a listener that may accept the packet of data from the remote entity for determining if the state information for the remote entity should be created in the second table of unverified remote entities.~~

23. (Currently Amended) The computer program product of claim 22, wherein [[if]]when the state information for the remote entity exists in the second table of unverified remote entities, but the remote entity cannot be verified, the method further comprises the act of:

checking if the packet includes a synchronization message for determining how to respond to the unverified remote entity.

24. (Original) The computer program product of the claim 23, wherein if the packet of data includes a synchronization message, the local server responds by either sending a synchronization-acknowledgement packet or by deleting the packet.

25. (Previously Presented) The computer program product of the claim 23, wherein if the packet of data does not include a synchronization message, the local server responds by one or

more of the following: deleting the packet, retransmitting the original message to the remote entity or removing the state information from the second table of unverified remote entities.

26. (Original) The computer program product of claim 22, wherein the first hash function is also a cryptographically secured hash function, and wherein the first and second hash functions are one of hardware based or software based.

27. (Currently Amended) The computer program product of claim 22, wherein [[if]]when state information for the remote entity does not exist in either the first table of verified remote entities or the second table of unverified remote entities, and wherein the server is a listener that may accept the package of data from the remote entity, the method further comprising the acts of:

creating state information for the remote entity within the second table of unverified remote entities; and

sending a synchronization-acknowledgement packet that includes an initial sequence number to the remote entity.

28. (Currently Amended) The computer program product of claim 22, wherein [[if]]when state information for the remote entity does not exist in either the first table of verified remote entities or the second table of unverified remote entities, and the server is not a listener that may accept the package of data from the remote entity, the method further comprises the act of:

sending a reset command to the remote entity for indicating that the packet was not verifiable and needs to be resent.

29. (Currently Amended) For a local server that receives data from one or more remote entities over a data transport protocol, a computer program product comprising computer readable storage media storing computer executable instructions that implement the method recited in claim 12. ~~a method of applying a cryptographically secure hash to packets from unverified remote entities for preventing denial of service attacks on lookup tables used to store state information for one or more remote entities, while maintaining the performance of the local server for packets from verified remote entities, the method comprising:~~

~~an act of receiving a packet of data from a remote entity that includes connection identifier information;~~

~~a step for determining if state information exists for the remote entity in a first table of verified remote entities;~~

~~if the state information for the remote entity does not exist in the first table of verified remote entities, a step for determining if state information exists for the remote entity in a second table of unverified remote entities;~~

~~if the state information exists in the second table of unverified remote entities, a step for determining if the remote entity can be verified such that state information can be moved to the first table of verified remote entities;~~

~~if state information does not exist in the second table of unverified remote entities; a step for determining if state information for the remote entity should be created in the second table of unverified remote entities.~~

30. (Currently Amended) The computer program product of claim 29, wherein [[if]]when the state information exists in the second table of unverified remote entities, but the remote entity cannot be verified, the method further comprises the act of:

checking if the packet includes a synchronization message for determining how to respond to the unverified remote entity.

31. (Original) The computer program product of the claim 30, wherein if the packet of data includes a synchronization message, the local server responds by either sending a synchronization-acknowledgement packet or by deleting the packet.

32. (Previously Presented) The computer program product of the claim 30, wherein if the packet of data does not include a synchronization message, the local server responds by one or more of the following: deleting the packet, retransmitting the original message to the remote entity or removing the state information from the second table of unverified remote entities.

33. (Cancelled)

34. (Original) The computer program product of claim 33, wherein the first hash function is also a cryptographically secured hash function, and wherein the first and second hash functions are one of either hardware based or software based.

35. (Currently Amended) The computer program product of claim 29, wherein [[if]]when state information does not exist in either the first table of verified remote entities or the second table of unverified remote entities, and wherein the step for determining [[if]]when state information for the remote entity should be created in the second table of unverified remote entities includes the act of checking whether the local server is a listener that may accept the package of data from the remote entity, the method further comprising the acts of:

creating state information for the remote entity within the second table of unverified remote entities; and

sending a synchronization-acknowledgement packet that includes an initial sequence number to the remote entity.

36. (Currently Amended) The computer program product of claim 29, wherein [[if]]when state information does not exist in either the first table of verified remote entities or the second table of unverified remote entities, the step for determining if state information for the remote entity should be created in the second table of unverified remote entities includes the act of checking whether the local server is a listener that may accept the package of data from the remote entity, and wherein the server is not a listener that may accept the package of data from the remote entity, the method further comprises the act of:

sending a reset command to the remote entity for indicating that the packet was not verifiable and needs to be resent.

37. (Original) The computer program product of claim 29, wherein the remote entity becomes verified by sharing a secret sent to the remote entity by the local server.